

DB2 and WebSphere - Capita Selecta

Dutch DB2 User Group (DDUG), 23.11.2006

Kris Van Thillo - ABIS Training & Consulting

Agenda

- **Introduction & Context**
- **A few words about WebSphere Application Server**
- **A few words about DB2 - which DB2?**
- **Java and DB2 interact!**
 - **SQLJ -> profiles**
- **Its about Connection Pooling**
- **Security and Authentication**
- **WebSphere, DB2 and Locking**
- **Dynamic SQL**
- **Java, SQLJ and DB2 - Performance issues**
- **Stick to your best practice guidelines...**

Introduction & Context - 1

So why are we interested in this subject?

- because ABIS decided to focus on z/OS based technologies, and technologies integrating with ...
- because ABIS decided to focus on the infrastructure required to develop applications in this environment, eg.
 - WebSphere Application Server
 - WebSphere MQSeries
- because it is about DB2
- because ABIS wants to at least implement - on a small scale - some of the technologies we are interested in!

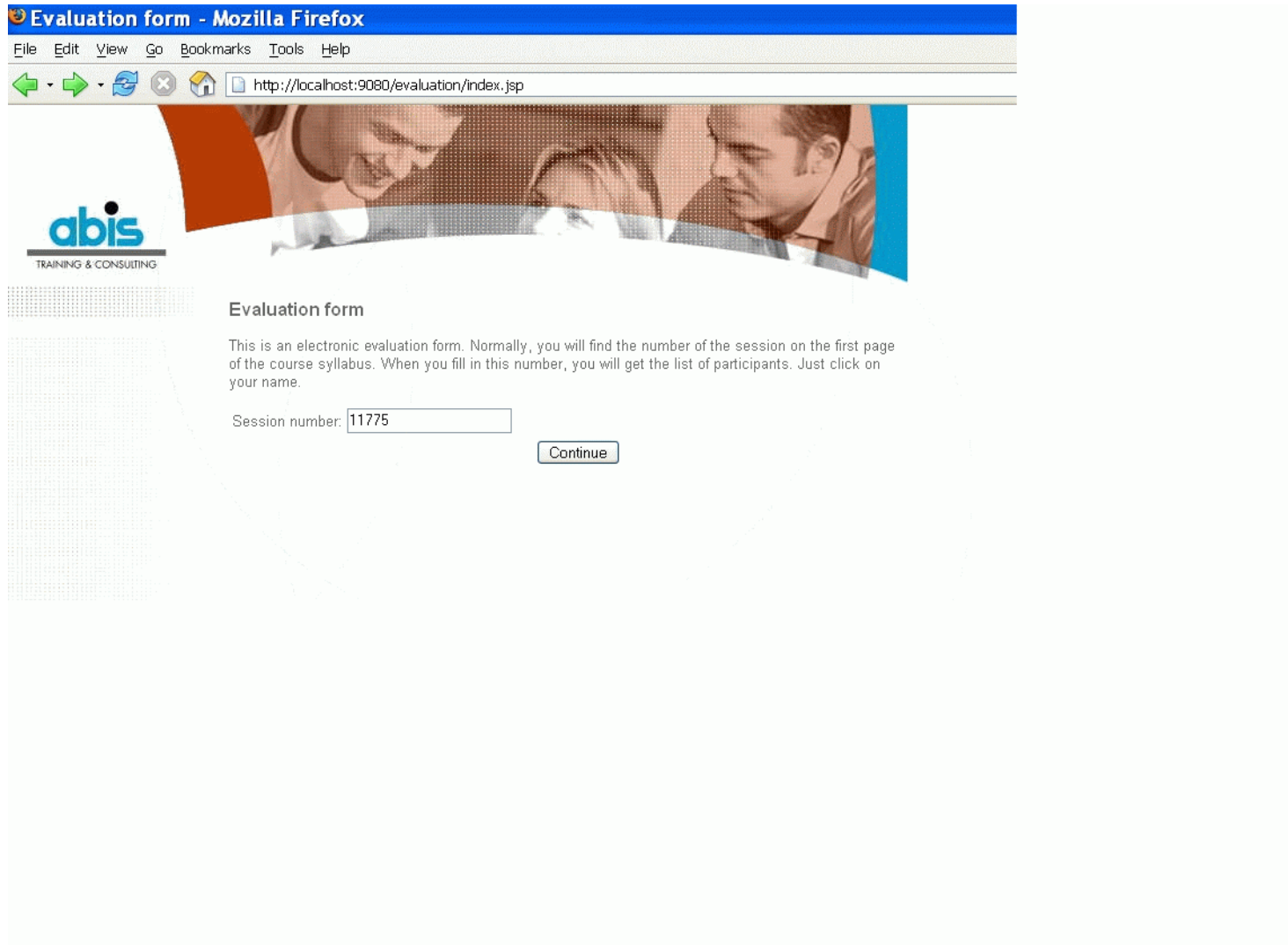
In this case: upgrading Course Evaluation Application!

Introduction & Context - 2 - old

If you want, you can alter these data
If you have chosen the wrong name [Click here](#)

Personal Data	Company Data
<ul style="list-style-type: none">• Your name KOOPS• Your first name Ronald• Your direct phone number at work <input type="text"/>• Additional phone number (GSM/06) <input type="text"/>• Personal email address at work <input type="text"/>• Your function <input type="text"/>• Your language Nederlands• How would you like to receive Abis information? <input type="radio"/> Preferably by email <input type="radio"/> Preferably by post• Would you like to receive the course calendar (twice a year) <input type="radio"/> Yes <input type="radio"/> No, thank you	<ul style="list-style-type: none">• Your company name BELASTINGDIENST - B/CICT• Company address JOHN F. KENNEDYLAAN <input type="text"/> 8 <input type="text"/> 7314 PS APELDOORN• P.O. Box POSTBUS 9050 7300 GA• Company telephone number 055-5281000• Fax number 055-5287114• General company email address (eg:info@company.com) <input type="text"/>• Web site www.belastingdienst.nl

Introduction & Context - 3 - new

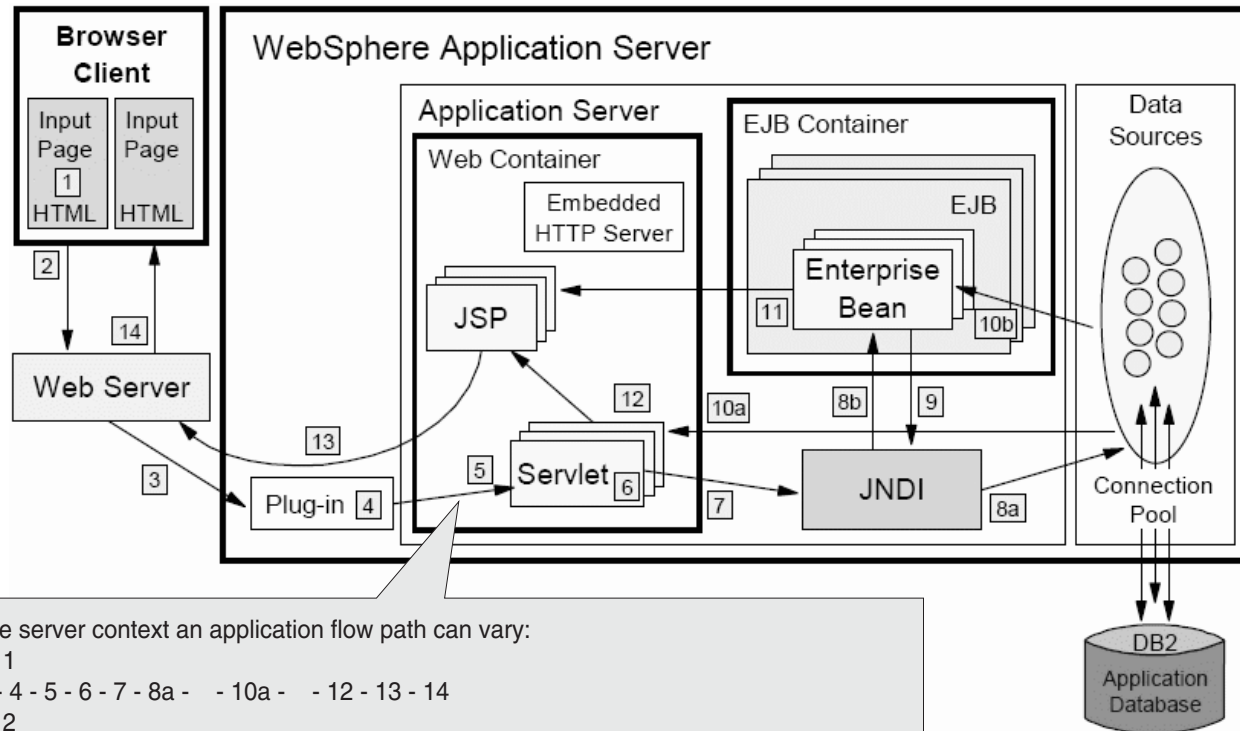


WebSphere Application Server - 1

WebSphere Application Server (WAS)

- **IBM's J2EE application platform**
 - available on a number of platforms
 - operates as a 'plugin' to the webserver, with HTTP requests being forwarded to WAS!
- **WAS for z/OS - the J2EE application platform for zSeries**
- **A J2EE application platform?**
 - a runtime 'container' for J2EE 'enterprise' applications
 - support for J2EE standard technologies(JDBC, RMI, JNDI, EJB, ...)
 - provides transaction management capabilities

WebSphere Application Server - 2 - the flow



Within the server context an application flow path can vary:

scenario 1
 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8a - - 10a - - 12 - 13 - 14

scenario 2
 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8b - 9 - 10b - 11 - - 13 - 14

even a scenario that bypasses the "enterprise business" part completely, is viable
 1 - 2 - 3 - 4 - 5 - 6 - Application DataBase - 12 - 13 - 14

WebSphere Application Server - 3 - the flow explained

1. A Web client requests a URL in the browser input page.
2. The request is routed to the Web server over the Internet.
3. All requests go to the WebSphere plug-in first.
4. The Web server plug-in examines the URL, verifies the list of host name aliases from which it will accept traffic based on the virtual host information, and chooses a server.
5. A stream is created. A stream is a connection to the Web container. It is possible to maintain a stream over a number of requests. The Web container receives the request and, based on the URL, dispatches it to the proper servlet.
6. If the servlet class is not loaded, the dynamic class loader loads the servlet (servlet init(), then doGet() or doPost()).
7. JNDI is used for lookup of either datasources or EJBs required by the servlet.
8. Depending upon whether a datasource/EJB is specified/requested, the JNDI directs the servlet:
 - 8a - To the corresponding database and gets a connection from its connection pool in the case of a data source.
 - 8b - To the corresponding EJB container, which then instantiates the EJB when an EJB is requested.
9. If the EJB requested involves SQL, it goes back to the JNDI to look up the datasource.
10. SQL is executed, data retrieved is sent back either to the servlet (10a) or to the EJB (10b).
11. Data beans are created and handed off to JSPs in the case of EJBs.
12. The servlet sends data to a JSP.
13. The JSP generates HTML that is sent back through the WebSphere plug-in to the Web server.
14. The Web server sends the output HTML page to the browser.

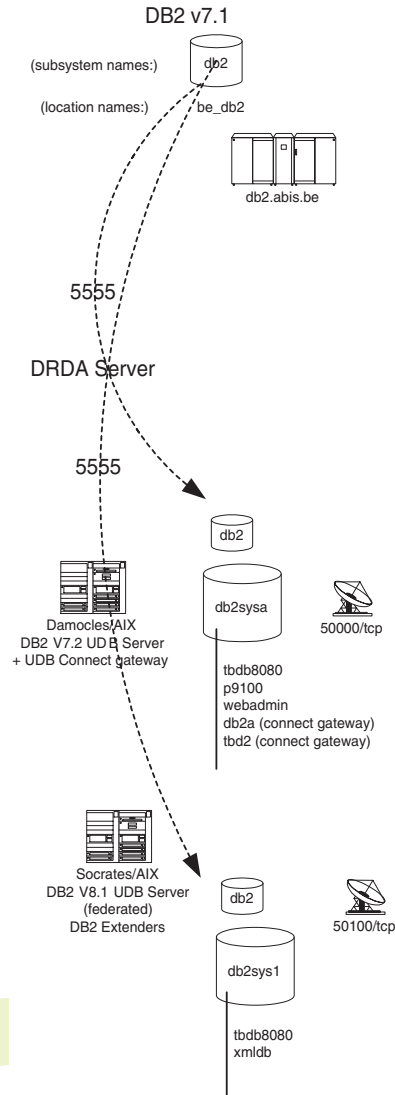
Which DB2? - 1

When considering DB2, it is important to take into consideration

- which DB2 we are talking about...
- HOW DB2 will be accessed:
 - locally or remotely - DB2 is AR and/or AS
 - through DB2 Connect - Connect is the AR; DB2 is the AS

Configuration/tuning options differ!

Which DB2? - 2



Java and DB2 interact - 1 - SQLJ vs JDBC

	SQLJ	JDBC
SQL	dynamic static (profiles!)	dynamic
ADevCycle	'traditional' ADC	no ADC - language extension!
Strong Typing	supported - DTs checked during ADC	not supported
Security checking	static	dynamic (runtime)
Standards based	+/- ANSI standard	SUN
Portable	Yes	Yes
	** best performance ** best authorisation	** most flexible

IBM DB2 Universal Driver for JDBC and SQLJ	
JCC	<ul style="list-style-type: none"> o) single driver across platforms o) common behaviour o) identical program preparation o) write anywhere, deploy anywhere

Java and DB2 interact - 2 - some remarks about SQLJ

- **adc(SQL) = adc (cobol)**
 - SQLJ commands extracted from source
 - bound into package for static SQL execution
 - 4 packages are created by default (RR, RS, CS and UR)
 - > optionally request for just one to be created
 - > only one 'is used'
- **dsnpc = sqlj translator**
 - > db2sqljcustomize command for STATIC package creation
 - > if not used: dynamic SQL
 - > tailor sqljproperties for warnings etc
 - > passwords; USS

Java and DB2 interact - 3 - Connectivity types

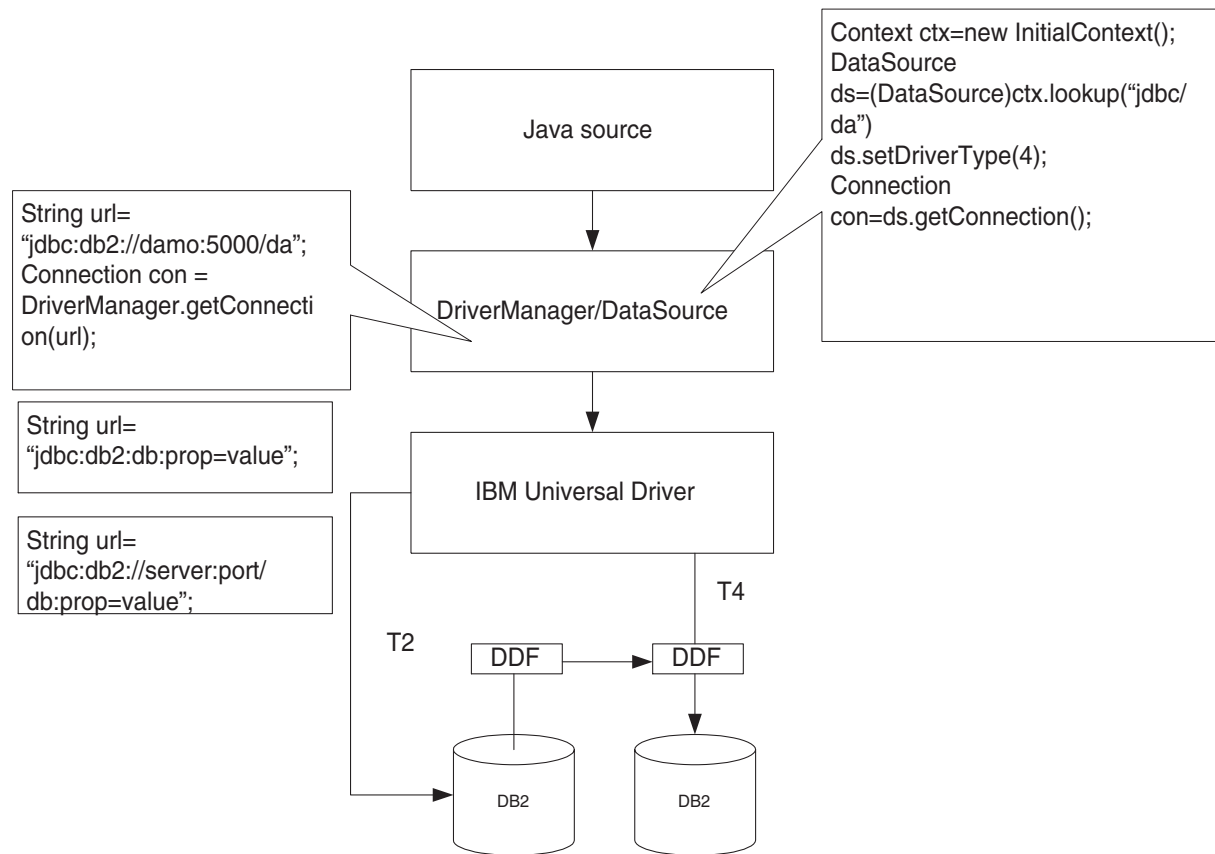
Type	Characteristics	Status
1	JDBC calls mapped to API/ODBC calls (client-side libs) - JDBC/ODBC bridge poor performance (2 APIs)	obsolete
2	Java accesses client-side libraries, required to access 'remote' databases - efficient z/OS: local connections (same LPAR) only	popular
3	100% pure Java Proprietary middleware required to translate calls to DB2 specific format Communicate with a specific server daemon running on the database - no 'client side' installation required	obsolete
4	100% pure Java - no 'client side' installation required Access to DB2 using open DB2 (DRDA RA) protocol	preferred

IBM DB2 Universal Driver for JDBC and SQLJ	
JCC	<ul style="list-style-type: none"> o) Type 2 and Type 4 supported o) single instance loaded o) concurrent use of o) concentrator function

Java and DB2 interact - 4 - Connection interfaces

DriverManager interface	DataSource interface
<pre>String url= "jdbc:db2://damocles:5000/da"; Connection con= DriverManager.getConnection(url) ;</pre>	<pre>Context ctx=new InitialContext(); DataSource ds=(DataSource)ctx.lookup("jdbc/da") ds.setDriverType(4); Connection con=ds.getConnection();</pre>
<ul style="list-style-type: none"> °) connection string explicitly specified °) minimal setup, minimal admin °) portability? -> hard coded string! °) security? -> hard coded + visible 	<ul style="list-style-type: none"> °) data source specified externally (defined by WAS administrator?) °) setup 'overhead' °) portability increases -> data source object °) security improved -> data source object
<p>T2 connection string</p> <pre>String url= "jdbc:db2:db:prop=value";</pre>	<p>DriverType property defines Type 2 or Type 4 connection type</p>
<p>T4 connection string</p> <pre>String url= "jdbc:db2://server:port/db:prop= value";</pre>	

Java and DB2 interact - 5 - types and interfaces

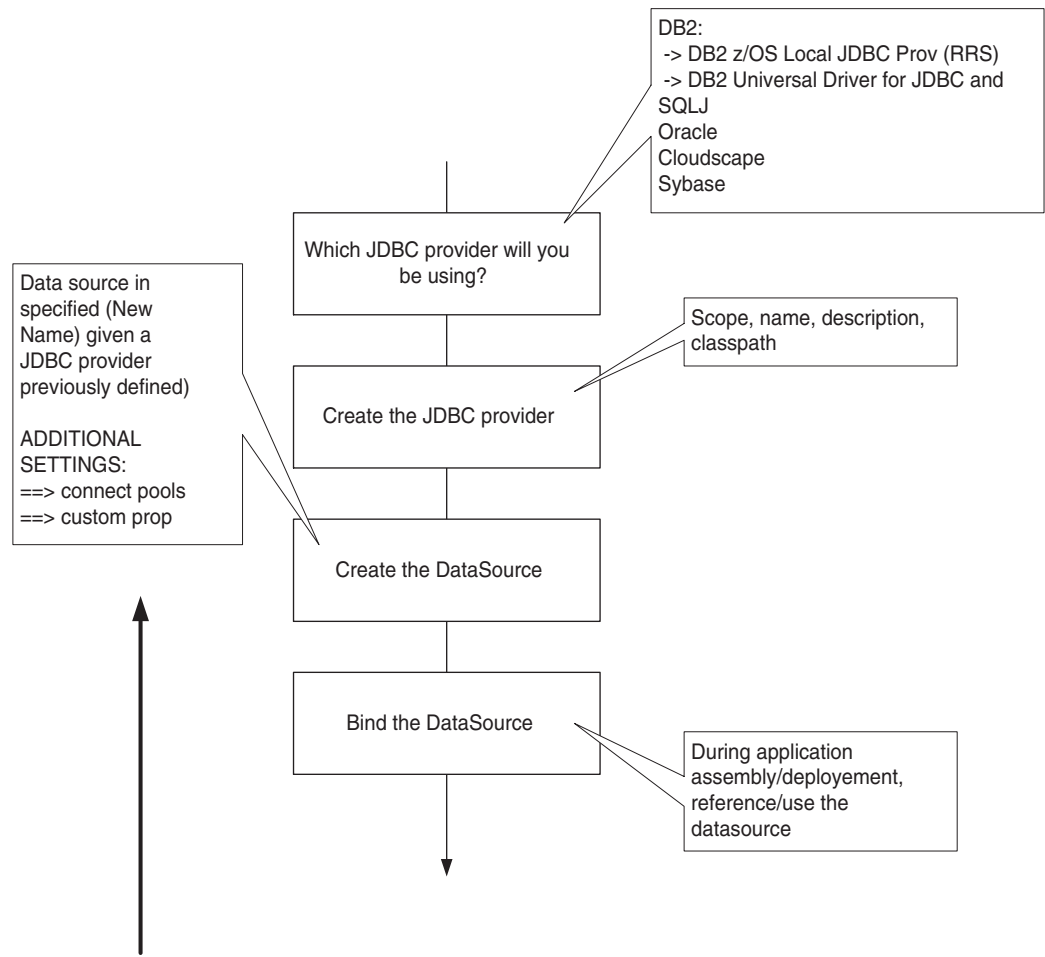


Java and DB2 interact - 6 - data access

From the WAS point-of-view, its all about:

- JDBC providers and data sources
- or
- Java 2 Connector connection factories

Java and DB2 interact - 7 - data sources

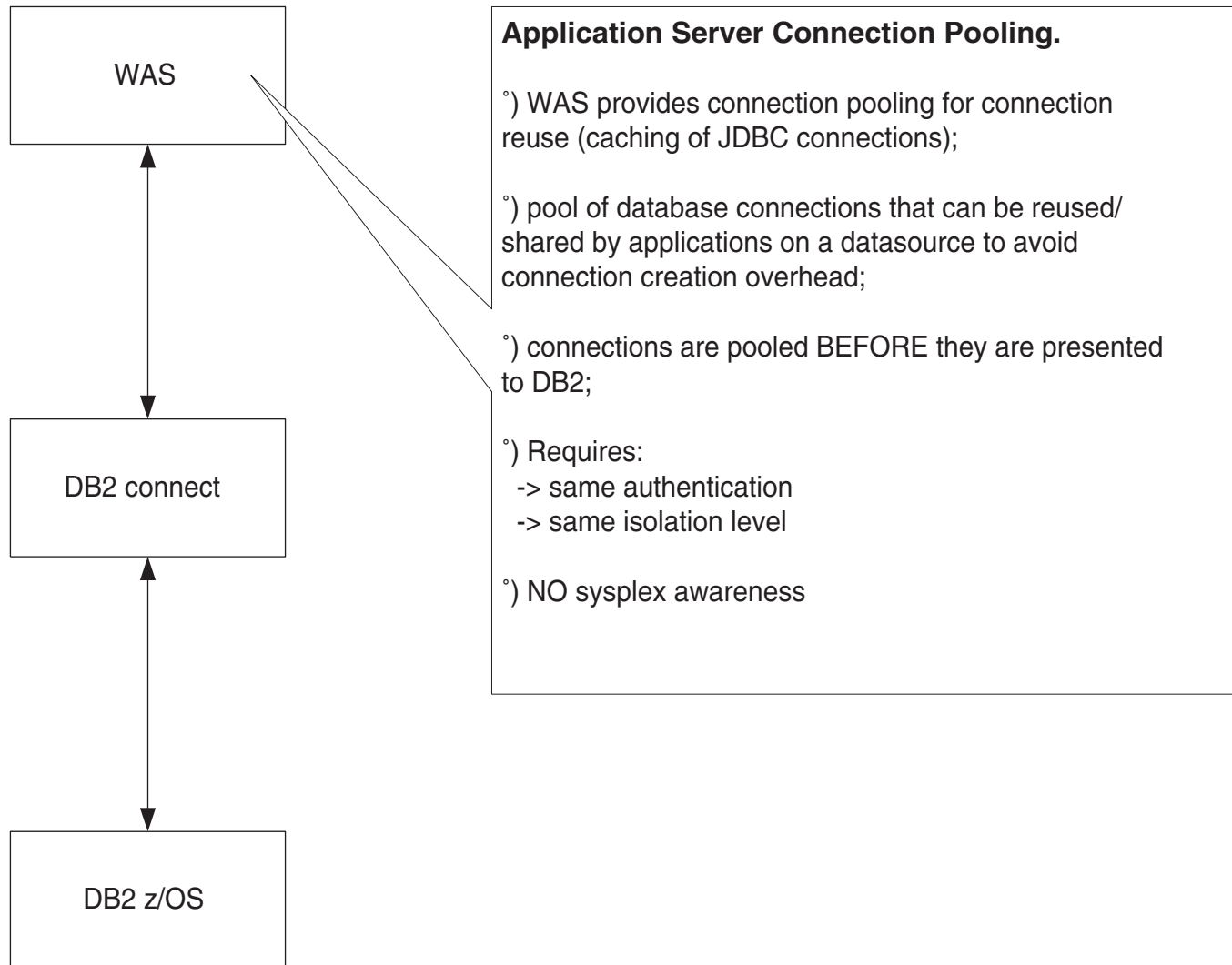


WAS connection pooling - 1 - introduction

Aim of Pooling:

- avoid waste of CPU cycles
- avoid waste of memory
- avoid waste of virtual storage

WAS connection pooling - 2 - implementation



WAS connection pooling - 3 - procedure

Procedure:

- **pool starts with zero connections, grows and shrinks number of connections;**
- **when a minimum number of connections is reached, it will not shrink below this minimum;**
- **the actual number of connections increases;**
 - maximum number can be defined
 - if max + 1 requested, WAS will wait for free connection a specific period of time
 - surplus connections > minimum number connections is closed automatically when idle a specified amount of time
 - connections can be stolen when not active a specific amount of time
- **connections can be shared:**
 - authentication
 - isolation
 - transaction

WAS connection pooling - 4 properties

Properties:

- **Max Connections**
- **Min Connections**
- **Reap Time** : auto cleaning
- **Connection Timeout**
- **Aged Timeout**
- **Unused Timeout**
- **Purge Policy**: EntirePool, FailingConnectionOnly
- **Statement Cache Size**

WAS connection pooling - 5 - exceptions

Consider these exceptions:

- **ConnectionWaitTimeoutException**
 - application has waited too long for a connections....
 - max reached?
 - max reached and not shareable?
- **StaleConnectionException**
 - probably a database issue...
 - database not started
 - database error caused the connection to be stale
 - error thrown because a JDBC resource references a stale connection
 - less likely when connections are new

WAS connection pooling - 6 - the cache

A quick word about the statement cache:

- allocated for the entire pool!
- used on a 'connection per connection' basis!

WAS Statement cache - 1 - JDBC statement types

- **Statement**

 - executes any sql string**

 - Statement stm = con.createStatement();
 - ResultSet rs = stm.executeQuery("SELECT * from TUTCOURSES where CDUR = 10");

- **PreparedStatement**

 - PreparedStatement stm = con.prepareStatement("UPDATE TUTCOURSES SET caprice = caprice * ? WHERE ctype = ?");
 - for(int i=1; i<10; i++)
 - {
 - stm.setInt(1, array[i][0]);
 - stm.setInt(2, array[i][1]);
 - stm.execute();
 - }

 - **substitution parameters allowed in the string**
 - **separates preparation/compilation and execution**
 - **using PreparedStatement objects**

- **A CallableStatement**

 - stored procedure based**

WAS Statement cache - 2 - procedure

- **application issues a prepare:**
 - **WAS looks through the statementcache for the connection to verify if a PreparedStatement object already exists:**
 - if it exists, it is returned to the application
 - if it does not exist, WAS requests DB2 for a prepare
 - **DB2 performs a lookup in its global statement cache, to verify if a prepared version of the statement already exists:**
 - if it exists, it is returned to WAS
 - if it does not exist, it is created
 - **DB2 will prepare a statement even if handed a PreparedStatement object from WAS if does no longer exist in the DB2 global statement cache**
 - **the statement is executed, and data returned to the application**
- **when closing/committing the application, the PreparedStatement object is cached in the WAS statement cache**
 - cache is managed FIFO
 - objects kicked-out if cache limit reached

WAS Statement cache - 3 - what about DB2?

- **DB2 UDB is the target?**
 - **database configuration parameter PCKCACHESZ, used for both static and dynamic SQL statements**
 - **elements are kept in cache as long as 'possible'**
 - **reuse possible beyond the scope of WAS**
 - **identical!**
- **DB2 UDB z/OS is the target?**
 - **DSNZPARM CAHEDYN YES**
 - **MAXKEEPD, EDMPOOL, EDMSTMTC**
 - **identical!**

DB2, WAS and Security - 1

Approaches:

- specify userid/password within application
- specify default user / password in data source
- allow for prompting

Observe!

- connection pooling depends on identical auth id for all threads to be shared
- better to have all web-user share one auth id - but how to distinguish between users?

'client strings' can be set using new API

ClientWorkstation, ClientUser, ClientProgramName, ClientAccountingInformation

SQLSETI API

- available in Universal Driver
- JDBC properties in the JDBC provider
- the same literal values are set to all connections to the data source from the WAS server
- fields are set at the connection level!

Dynamic SQL - 1 - avoiding it!

- **DynamicRules(BIND)**
- **SQLJ**
- **StoredProcs**
- **Static profiling?**

Dynamic SQL - 2 - Static Profiling

- **Allows dynamic SQL to be captured from dynamic SQL programs and bound into packages**
 - db2cli.ini -> STATICMODE = CAPTURE**
 - db2cap utility**
 - enable statement matching in db2cli.ini**
- **runtime: SQL statements are substituted at the DB2 client with static SQL calls**
- **Requirements:**
 - **CLI!**
 - **text must of course be identical**
- **Why use it?**
 - **security?**
 - **performance?**

WebSphere, DB2 and Locking - 1

JDBC/SQL can dynamically set isolation levels:

- attribute of the database connection
- set programmatically or as part of the datasource definition

Connection con.setTransactionIsolation()

in SQLJ:

SET TRANSACTION ISOLATION LEVEL

isolation level mapping:

DB2

Uncommitted read (UR)

Cursor stability (CS)

Read stability (RS)

Repeatable read (RR)

JDBC/SQLJ/WAS

TRANSACTION_READ_UNCOMMITTED

TRANSACTION_READ_COMMITTED

TRANSACTION_REPEATABLE_READ

TRANSACTION_SERIALIZABLE

WebSphere, DB2 and Locking - 2

Order of precedence:

- **SQL statement, WITH option (UR, CS, RS, RR)**
- **last specified setTransactionIsolation()**
- **WebSphere datasource reference value**
- **Default for JDBC in WAS: RS**

Java and DB2 - Performance issues

- **'Always' use SQLJ instead of JDBC**
 - > **static SQL!**
- **Disable autoCommit**
DriverManager connections default = on
- **'select * from' : deadlier than ever!**
 - **unicode conversion**
 - **Java object manipulation**
- **Release JDBC resources**
 - **result sets, PreparedStatements, CallableStatements**
 - **you need to manually close iterators!**
- **JDBC batch updates - JDBC batch query?**
 - **multiple updates become ONE single command**
 - **single command = single network trip = better performance**
 - **errors?**

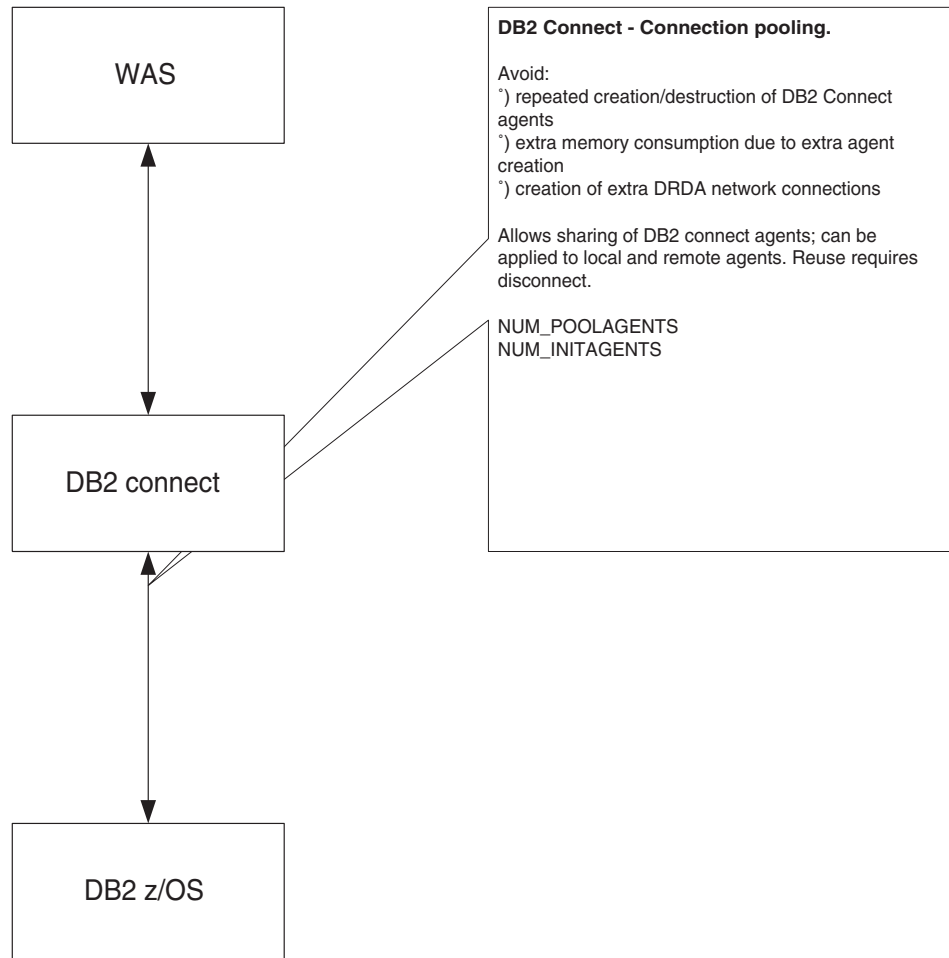
SQLJ - Performance issues

- **Missing/failed customisations**
- **Use positioned iterators**
- **Scrollable iterators**
- **Use matching data types**

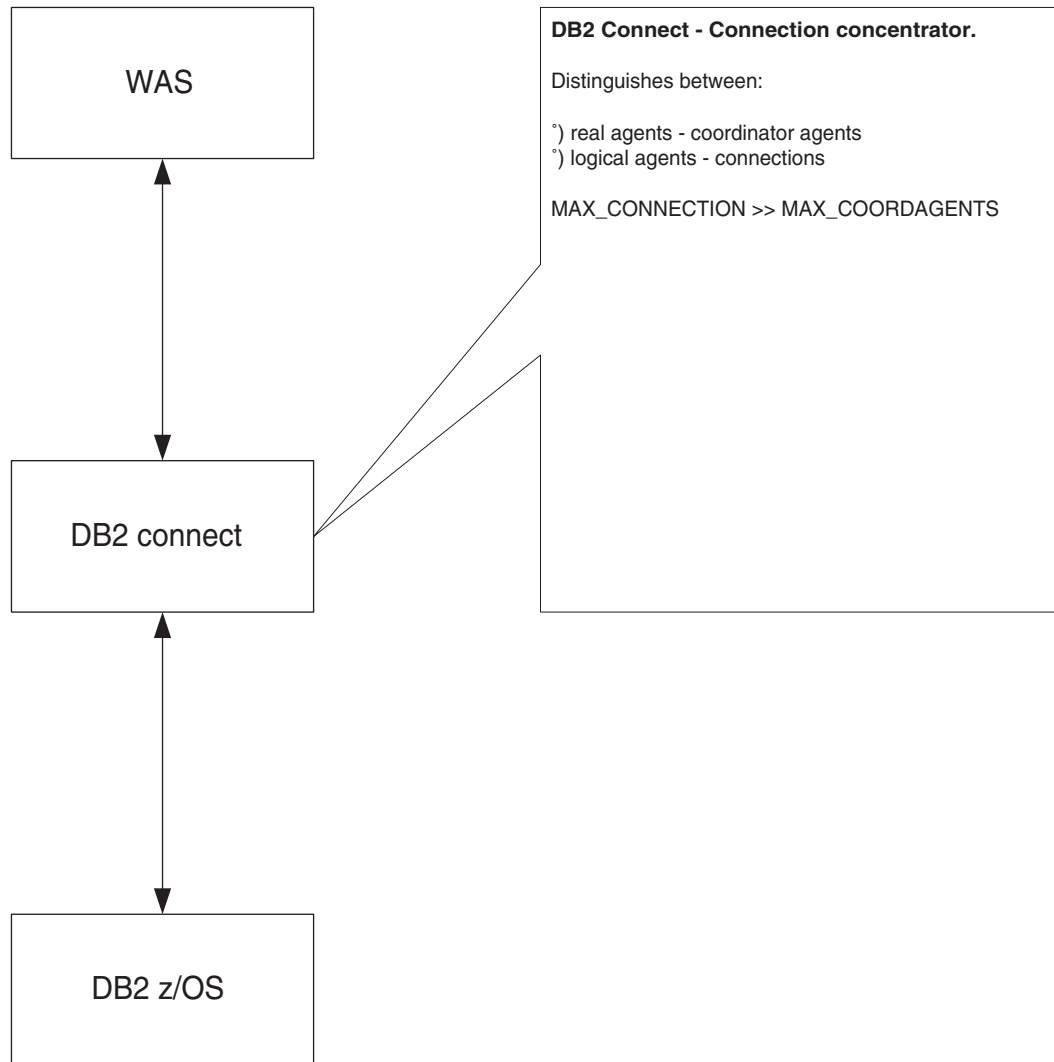
Best practice guidelines

- **stored procedures!?**
- **explain!**

Connection Pooling - DB2 Connect connection pooling



Connection Pooling - DB2 Connect connection concentrator



Q&A

Thank you!

Kris Van Thillo

kvanthillo@abis.be

abis

TRAINING & CONSULTING

Thank you!